The opinion in support of the decision being entered today was not written for publication and is not binding precedent of the Board.

Paper No. 29

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte FRANCIS X. MCKEEN, MICHAEL C. ADLER,
ROBERT P. NIX, JOEL S. EMER, DAVID J. SAGER, and
P. GEOFFREY LOWNEY

Appeal No. 1999-0495
Application No. 08/752,729

ON BRIEF

Before KRASS, BARRY, and BLANKENSHIP, Administrative Patent Judges.

BLANKENSHIP, Administrative Patent Judge.

DECISION ON APPEAL

This is a decision on appeal under 35 U.S.C. § 134 from the examiner's final rejection of claims 18-20.

We reverse.

BACKGROUND

The invention is directed to a method for executing computer instructions in parallel, with management of exception conditions that may occur. Exception conditions are signals used to indicate unexpected conditions during processing. Representative claim 18 is reproduced below.

> 18. A method for executing instruction in parallel in a computer system, comprising:
>
> identifying each instruction to be executed with an identification which is independent of an execution order;
>
> reordering the instructions independent of the identification of the instructions;
>
> executing the instructions in parallel as a plurality of execution flows, each instruction of each corresponding execution flow identified with the corresponding execution flow, the instructions of each execution flow executing in parallel with each other;
>
> determining if a particular instruction is subject to an exception condition;
>
> identifying a particular execution flow corresponding to the identification of the particular instruction; and
>
> repeating executing of the instructions of the particular execution flow to resolve the exception condition.

The examiner relies on the following references:

Kodama	5,287,466	Feb. 15, 1994 (filed Mar. 25, 1991)
Popescu et al. (Popescu)	5,561,776	Oct. 1, 1996 (effectively filed Dec. 5, 1990)

Michael D. Smith et al. (Smith), Boosting Beyond Static Scheduling in a Superscalar Processor, Computer Systems Laboratory, Stanford University, Stanford CA, pp. 344-354 (copyright 1990 IEEE).[1]

Claim 18 stands rejected under 35 U.S.C. § 102 as being anticipated by Popescu.

Claims 19 and 20 stand rejected under 35 U.S.C. § 103 as being unpatentable over Smith and Kodama.

Claims 1, 2, 4-11, and 13-17 have been allowed.

Claims 3 and 12 have been canceled.

We refer to the Final Rejection (mailed Nov. 13, 1997) and the Examiner's Answer (mailed Sep. 15, 1998) for a statement of the examiner's position and to the Brief (filed Jul. 16, 1998) for appellants' position with respect to the claims which stand rejected.

OPINION

The examiner's findings underlying the rejection of claim 18 as being anticipated by Popescu are set forth on page 3 of the Answer. Appellants advance several arguments in opposition to the rejection. We are persuaded by appellants that Popescu fails to disclose the step of "reordering the instructions independent of the identification of the instructions," and thus cannot support a rejection for anticipation.

[1] The paper was apparently published in a paginated IEEE journal, but the copy of record does not tell in which journal it appeared.

For the initial claim 18 step of "identifying each instruction," the rejection points to column 4, lines 49-61 of the reference. The section reveals that as each instruction is fetched from the cache, a counter assigns strictly sequential ID values to each instruction in the order fetched. A portion of the ID of a particular instruction may be compared to the corresponding portion of the ID of another instruction. The comparison provides the relative order of when the instructions were fetched from memory, and thus an "age" comparison between instructions.

For the contested step of "reordering," the rejection points to column 7, lines 52-65 of Popescu. The section describes an instruction scheduler.

> Among the information about instructions examined by the instruction scheduler to determine whether the instruction should be executed is the locker information and instruction ID. From this information, the instruction scheduler 33 picks the instructions most ready to run.... The instruction scheduler picks the oldest runnable instructions for which there are sufficient execution resources available.

In response to appellants' argument that the reordering is not "independent of the identification of the instructions," the examiner points to further material at column 6, lines 48-67 of the reference and submits a rebuttal. (Answer at 5-6.) However, we do not find that the examiner's response is sufficient to demonstrate that what appears to be clear language in the patent, describing a process contrary to the clear language of instant claim 18, in actuality meets the requirements of the claim.

That all the details of Popescu's process are not shown in the drawings is not surprising; the written description of a disclosure is normally more detailed than the accompanying drawings. If we understand the examiner's position to be that Popescu's instruction ID is not the only factor, or even a factor of primary importance, in the "reordering," then we may agree with the finding, but do not consider such to meet the terms of claim 18. In particular, while we agree that the instruction ID does not "determine" an execution scheduling order for the instructions (Answer at 6), the examiner's observation might indicate an erroneous claim interpretation.

The instant specification (page 23) describes the compiler assigning an identification field 105 "independent of an execution order." The specification further describes (pages 13-14) the compiler reordering the instructions "in order to increase total system throughput," with no disclosure of the identification field 105 being considered in the reordering. Or, in the terms of claim 18, the reordering is "independent of the identification of the instructions."

“Anticipation is established only when a single prior art reference discloses, expressly or under principles of inherency, each and every element of a claimed invention.” RCA Corp. v. Applied Digital Data Sys., Inc., 730 F.2d 1440, 1444, 221 USPQ 385, 388 (Fed. Cir. 1984). In our opinion the reordering of the instructions, as disclosed by Popescu," is not "independent of the identification of the instructions." We therefore cannot sustain the section 102 rejection of claim 18.

We next consider the rejection of claims 19-20 under section 103 as being unpatentable over Smith in view of Kodama, set forth on pages 3 through 4 of the Answer. Appellants argue (Brief at 16-17) that, contrary to the statement of the rejection, Smith fails to disclose the first step of claim 19: "identifying each instruction to be executed with an identification which is independent of an execution order."

We agree with appellants that the identification accorded by the "boosting bit," as disclosed by Smith, cannot fairly be considered "independent of an execution order." We may agree with the examiner (Answer at 6) that the bit does not necessarily indicate the actual order of execution in the system. However, whether or not the bit is set "depends upon the outcome of the next conditional branch." Smith at 347, col. 1, ll. 7-8. "Boosted instructions are conditionally committed upon the result of later branch instructions." Id. at 344, col. 2, ll. 30-31. Since the identifier is set upon consideration of branch instructions occurring later in execution, the identification is not "independent" of an execution order.

Additionally, the rejection over Smith in view of Kodama is unclear. The statement of the rejection, set forth on pages 3 and 4 of the Answer, purportedly corrects a typographic error present in the Final Rejection. (See Answer at 7.) However, in the Answer, Smith is relied upon as teaching "grouping instructions into a plurality of sets" and identifying each instruction. Further, "Kodama taught (e.g. see figs 1-6) grouping instructions into a plurality of sets according to identification (7) and reordering the instructions in the sets to accelerate execution (5a, 6a)." (Answer at 4.) Instant claim 19 requires "grouping instructions into a plurality of sets according to said identification." Whatever "identification" might be pointed out in Kodama, the identification is, manifestly, not the identification (i.e., boosting bit) disclosed by Smith. It is thus unclear which reference is relied upon for particular requirements of claim 19.

Finally, the rejection appears to contemplate that Smith's setting of the "boosting bit" is both "identifying each instruction" and a form of "grouping instructions into a plurality of sets" -- boosted or not boosted. While the binary form of identification might effectively group instructions into two sets, instant claim 19 requires separate steps of "identifying" and "grouping," as set forth in the initial portion of the claim. The rejection does not point out where any actual "grouping" of the boosted and non-boosted instructions is disclosed by Smith. We do not find disclosure or suggestion of the separate steps as claimed, even if setting of the boosting bits were to be considered "independent of an execution order."

For the foregoing reasons we cannot sustain the section 103 rejection of claims 19 and 20 as being unpatentable over Smith in view of Kodama. We thus do not sustain the rejection of any of the claims on appeal.

CONCLUSION

The rejection of claims 18-20 is reversed.

REVERSED

ERROL A. KRASS	)
Administrative Patent Judge	)
	)
	)
	)
	) BOARD OF PATENT
LANCE LEONARD BARRY	) APPEALS
Administrative Patent Judge	) AND
	) INTERFERENCES
	)
	)
	)
HOWARD B. BLANKENSHIP	)
Administrative Patent Judge	)

Jonathan M. Harris
Conley, Rose, & Tayon
P.O. Box 3267
Houston, Texas 77253-3267